(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 926 863 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
30.06.1999 Bulletin 1999/26

(51) Int Cl.[6]: **H04L 29/06**

(21) Application number: **98309972.2**

(22) Date of filing: **04.12.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **16.12.1997 US 991943**

(71) Applicant: **Hewlett-Packard Company**
Palo Alto, California 94304 (US)

(72) Inventors:
- **Kimball, Karen E.**
  Sacramento, California 95833 (US)
- **Faulk, Robert L. Jnr.**
  Roseville, California 95747 (US)
- **Maguire, Robert M.**
  Roseville, California 95747 (US)

(74) Representative: **Powell, Stephen David et al**
WILLIAMS, POWELL & ASSOCIATES
4 St Paul's Churchyard
London EC4M 8AY (GB)

(54) Automatic configuration of network links

(57) An intelligent network agent (12) software application for multi-segment network devices (such as bridges or multi-segment repeaters) which have one or more network links automatically configures the network links. The configuration includes: assigning bridged links to different segments (to maximize the connectivity offered by the bridging and to prevent looping); setting up bridged links on a bridge to form full connectivity across the network with another bridge; setting up redundant connections; disabling ports which are causing loops in the network, or other configuration actions. The agent (12) is preferably fully embedded in the hardware device. However, the agent can also exist remotely and communicate with other software intelligence embedded in the hardware device via either an in-band (network) or an out-of-band (non-network) hardware connection.

EP 0 926 863 A2

Printed by Jouve, 75001 PARIS (FR)

Description

[0001] The invention relates to networked computer systems. More particularly, the invention relates to a method and system for automatically configuring multi-segment network device connections.

[0002] Computer network systems are inherently complex. A significant amount of technical education is required to understand, configure, and maintain any particular type of physical network, such as 10Base-T, Ethernet, Token Ring, or Token Bus networks. As the use of computer networks becomes widespread, new technicians are needed to maintain and operate those networks. These technicians may have little or no knowledge about networks, and may not ever fully understand networks, even with extensive training.

[0003] As a result, software schemes which can configure, maintain, or improve network conditions without requiring the user to have much specific technical knowledge are desirable. "Ease-of-use" and "user-friendly" schemes are often included in software applications that are embedded in, or located external to, the network devices. Such software applications are known as "intelligent network agents." An example of an intelligent network agent is disclosed in the U.S. Patent Application entitled, "Method and System for Automatic Detection of Bridged and Repeated Network Device Connections." U.S. Serial No. 08/895,119, filed July 16, 1997, and assigned to Hewlett-Packard Company.

[0004] The different types of network device connections between network links, along with their location in relation to a given hardware environment, are collectively known as a network's "topology."

[0005] For purposes of the discussion herein, a "repeater" is a device that is used to extend the length, topology, or interconnectivity of a physical cabling medium beyond that imposed by a single cable. Repeaters perform the basic actions of restoring the signal amplitude, waveform, and timing applied to the data signals.

[0006] For purposes of the discussion herein, a "bridge" is a multiport device connecting two or more network segments. When two stations on different network segments communicate with each other, the bridge forwards the packets between the two segments. When the stations are on the same segment, the bridge does not forward the packet to any other segment. The term "switch" can also be used interchangeably with the term "bridge."

[0007] One challenge facing a network administrator is correctly setting up a network. Setting up a network involves connecting devices together in such a way that they function as the administrator desires. An administrator who does not fully understand the workings of the devices, or the network itself, may not be able to set up the network properly. Unfortunately, there are no known satisfactory solutions to automatic configuration of external network device connections

[0008] Physical connections in the network can create a loop. When such a loop occurs, it is very difficult or even impossible to exchange information reliably with other network software agents. Some network software agents may cease working altogether. Manually reconfiguring a network to remove loops requires a certain degree of skill, and can be time consuming.

[0009] A failure in a network link can cause loss of valuable data. It can therefore be desirable to provide a redundant connection to insure data continuity in the event of such link failure. The only known method for automatically removing a loop or automatically configuring such redundant connections is the Spanning Tree Protocol described in IEEE standard 802.1. However, this method cannot be implemented using repeaters.

[0010] It would therefore be an advantage to provide a new method and system for automatically configuring network links to attached devices which can be implemented with both bridging devices and repeaters. It would be a further advantage if such method and system also created redundant connections for increased device reliability, fixed network loops, and otherwise configured the devices to work together. It would be yet another advantage if such method and system were user-friendly and required no specific action on the part of the network administrator.

[0011] The invention is an "intelligent network agent" software application for use in configuring multi-segment network devices (such as bridges or multi-segment repeaters) which have one or more network links. The invention encompasses several automatic configuration and loop-recovery features, each of which apply toward intelligently making decisions on the user's behalf, without special knowledge or intervention on the part of the user.

[0012] Such configuration could include: assigning bridged links to different segments (to maximize the connectivity offered by the bridging and to prevent looping), setting up bridged links on a bridge to form full connectivity across the network, setting up redundant connections, disabling ports which are causing loops in the network, or other configuration actions

[0013] In the preferred embodiment, the agent is fully embedded in that hardware device. However, in an alternative embodiment, the agent exists remotely and communicates with other software intelligence embedded in the hardware device via either an in-band (network) or an out-of-band (non-network) hardware connection. This out-of-band communication can remain operational even when loops are present in the network. While the target device has some embedded software intelligence in this alternative embodiment, the agent and thus the invention itself is not embedded.

Fig 1 is a diagram showing an embedded intelligent network agent according to the invention;

Fig. 2 is a diagram showing an intelligent network agent located external to the hardware environment;

Fig. 3 is a diagram showing a first sample topology for a hardware environment according to the invention;

Fig. 4 is a chart of the types of connections in the first sample topology according to the invention;

Fig. 5 is a flowchart of the algorithm used to automatically configure external network device connections according to a preferred embodiment of the invention;

Fig. 6 is a diagram of a configured hardware environment of that first sample topology according to the invention;

Fig. 7 is a diagram showing a second sample topology for a hardware environment according to the invention;

Fig. 8 is a diagram of a configured hardware environment of the second sample topology according to the invention.

[0014] The invention is an "intelligent network agent" software application for use in configuring multi-segment network devices (such as bridges or multi-segment repeaters) which have one or more network links.

[0015] The invention uses information regarding what kinds of devices are directly connected to the network links of a given hardware environment. This information can be automatically detected by the intelligent agent, furnished by another embedded or external application, or supplied by a user. The method for supplying such information does not affect the operation of the invention.

[0016] This information can then be used by the invention to automatically fix network problems, configure the network links, and/or offer users new "ease-of-use" features, or for other intelligent applications. Such configuration could include: assigning bridged links to different segments (to maximize the connectivity offered by the bridging and to prevent looping); setting up bridged links on a bridge to form full connectivity across the network; setting up redundant connections: disabling ports which are causing loops in the network; or other configuration actions. The invention encompasses several automatic configuration and loop-recovery features, each of which apply toward intelligently making decisions on the user's behalf, without special knowledge or intervention on the part of the user.

[0017] The invention is readily used with existing networks and devices, *i.e.* devices attached (either directly or indirectly) to the target device's network hardware environment may be of any known general purpose type and do not require special hardware or software to allow implementation of the invention.

[0018] The intelligent network agent can be embedded in the hardware environment, or can be an external software application. Fig. 1 is a diagram showing an embedded intelligent network agent according to the invention. Repeaters 1 through N 16, 18, 20, 22, together form the repeater bus 14. Each target network device has associated Ports 1 through N, respectively 24, 26, 28, 30. In this first example, an intelligent agent software module 12 is fully embedded in the hardware environment 10 and is in direct communication with the repeater bus.

[0019] Fig. 2 is a diagram showing an intelligent network agent located external to the hardware environment 32. In one embodiment, the intelligent network agent communicates with the repeater bus 14 through a port 26, using an in-band (network) hardware control connection 34. In another embodiment, the intelligent network agent communicates with the hardware environment through an out-of-band (non-network), hardware control connection 36. Such out-of-band communication can remain operational even when loops are present in the network.

[0020] Given information noting how the hardware environment's network links are connected to one another, *i.e.* whether they are repeated, bridged, or not connected, the intelligent agent can automatically configure the environment such that bridging and repeating will work without creating loops. Fig. 3 is a diagram showing a first sample topology for a hardware environment according to the invention.

[0021] The sample agent hardware environment 40 is a four-segment repeater with twelve network links L1-L12. In Fig. 3, the intelligent network agent software 42 is fully embedded in the target device's hardware environment. However, the following discussion is equally applicable to an external agent and thus is provided for purposes of example and not as a limitation on the scope of the invention. Sets of repeating devices R2, R7, R9 and bridging devices B4, B11, B12 are directly connected to the target device's hardware environment.

[0022] Fig. 4 is a chart of the types of connections in the first sample topology according to the invention. In the chart, ConnToRptrSet 44 indicates if there is a repeating device between any two links. Similarly, ConnToBridgeSet 46 indicates if there is a bridging device between any two links. This information denotes precisely, from the target device's hardware environment view, what kind of device connection(s) attaches each link to the rest of the remaining links. Therefore, this information approximates the type of devices attached to each of the target device's links.

[0023] Links associated with a RepeaterSet are connected to one or more individual repeaters which collectively repeat data as a single logical repeater, and, for the purposes of this invention, are treated as a single repeater. For example, repeater R9 of Fig. 3 could actually consist of two repeaters, but is treated by this invention as a single

repeater. Likewise, links associated with a BridgingSet are connected to one or more individual bridges, which collectively behave as a single bridge. Bridges forward packets if the destination is not on the originating segment and they filter when the destination is on the originating segment. The function of a bridge is to limit packets such that they only go on those segments which are needed.

[0024] The RepeaterSets and BridgingSets shown in Fig. 4 "govern" the connections between the links (*i.e.*, they control the communications patterns). In Fig. 4, a comparison of links L1 through L12 has shown that repeating device R2 connects L1 to L2, while bridging device B4 is shown connecting L1 to L4-L7. Thus, while there may be other devices between these links in the target device's hardware environment that the agent is not able to detect, these devices do not determine the communication patterns between the links. For example, a repeating device or RepeaterSet located between Link5 and bridging device B4 would not affect communication between Link5 and other links in the hardware environment, and can be ignored for purposes of the invention.

[0025] When both a RepeaterSet and a BridgingSet are attached to the same link, the RepeaterSet takes precedence in governing the link. When a device set is attached only to a single link, such as in BridgingSet B12 in the Figure, this device set is not perceived by the hardware environment as affecting communications from that link, and is also ignored for the purposes of the invention.

[0026] Device distinction cannot be perceived beyond a BridgingSet. Thus, for Link1 of Fig. 4, both the RepeaterSet R2 and the BridgingSet B4 can be seen. However, the RepeaterSet R7 beyond BridgingSet B4 cannot be perceived and thus is ignored for purposes of the invention. Similarly, any connection beyond BridgingSet B4 cannot be perceived at either Link4 or Link5.

[0027] The invention uses an algorithm to configure such multi-segment target network devices having one or more network links. This algorithm can be used, for example, to ensure that any connection to a single bridging device is on its own segment. This algorithm makes use of bridging capabilities, while preventing the occurrence of loops.

[0028] Fig. 5 is a flowchart of the algorithm used to automatically configure external network device connections according to a preferred embodiment of the invention. For each link (100) in the network that has an attached BridgingDevice but no attached RepeatingDevice (105), it is first determined whether the BridgingDevice is the preferred ("Best") BridgingDevice.

[0029] The preferred Bridging Device can be a device about which an application or user has provided specific information. Such information can include whether the device is recognized by the network or the user while other devices are not recognized, that the device is faster, or that the device is otherwise preferred. In a multi-segment hardware environment, the preferred Bridging Device could be the device having the most connections

[0030] The preferred embodiment of the invention is used with a four-segment hardware environment. However, the invention is also operable with other hardware environment configurations.

[0031] For the preferred BridgingDevice (110), if segments are available on which to put the link (115), the link is put on a segment and enabled (120). If segments are not available, (125), the link is disabled (130).

[0032] For a BridgingDevice that is not the preferred BridgingDevice (135), if no other links to this extra BridgingDevice have been configured (140), the link is put on any supported segment and enabled (145). A supported segment is one for which connectivity to other segments is supported. For example, a segment can be supported because the hardware environment is single-segment, or because bridging from the given segment to other populated segments can occur. If other links of the extra BridgingDevice have been configured (150), the link is disabled (155).

[0033] For a link that has an attached RepeatingDevice (160), if another link for the RepeatingDevice has already been configured (165), the link is disabled (170). If no other link for the RepeatingDevice has already been configured (175), and the preferred BridgingDevice is also attached to this link (180), it is then determined whether any segments are available on which to put the link. If segments are available (190), the link is put on a segment and enabled (195). If segments are not available (200), the link is disabled (205). If the preferred BridgingDevice is not attached to this link (210), the link is put on any supported segment and enabled (215).

[0034] The algorithm is further described in Table 1. Therefore, the data of Fig. 4 should be interpreted according to the pseudo-code listed in Table 1 below

TABLE 1

```
segCount = 0;

For each Link (using ConnToBridgeSet and ConnToRptrSet info)
(
 If any BridgingDevice attached AND no RepeatingDevice attached
      (
       if this is the "best" BridgingDevice in the hardware
                              environment
            (
             if there are still segments available to put link on
                  (
                   put link on segCount segment and enable link
                   segCount++
                  )
             else disable link
            )
       else    / *Not the "best" one* /
                  (
                   if no other links to this extra BridgingDevice have
                        been configured
                    put link on any "supported" segment & enable link
                   else disable link
                  )
      )
```

```
else if any RepeatingDevice attached
    if another link for this RepeatingDevice has already been
                                configured
     disable link
    else
            (
              if "best" BridgingDevice is also attached to this link
                    (
                     if there are still segments available to put link on
                            (
                              put link on segCount segment & enable link
                              segCount++
                            )
                      else disable link
                    )
              else put link on any "supported" segment & enable link
            )
    )
)
```

[0035] The "supported" segment, referred to in Table 1 (and discussed above), is a segment for which connectivity to other segments is supported. Connectivity is supported for such reasons as because the hardware environment is single-segment, or because bridging from the given segment to other populated segments can occur.

[0036] In the algorithm, any BridgingDevice is really a BridgingSet, and any Repeater is really a RepeaterSet. This algorithm is not a general purpose topology algorithm, in that individual bridges within a bridge set performing a bridging operation between a set of links connected to the target device are not identified. For example, in Fig. 4, BridgingSet B11 could in actuality be two or more separate bridges. However, for the purposes of the invention and this algorithm, they act as a single logical bridge. The same concept applies to repeaters.

[0037] Table 2 lists the actions resulting from the application, by the invention, of the Table 1 pseudo-code to the ConnToRptrSet and ConnToBridgeSet data of Fig. 4.

TABLE 2

Link	ConnToBridge	ConnToRptr	ACTION
1	B4	R2	Put on segCount segment, segCount++
2	B4	R2	Disable looping R2 Link2
3	-	-	--
4	B4	-	Put on segCount segment, segCount++
5	B4	-	Put on segCount segment, segCount++

TABLE 2 (continued)

Link	ConnToBridge	ConnToRptr	ACTION
6	B4	R7	Put on segCount segment, segCount++
7	B4	R7	Disable looping R7 Link7
8	-	R9	--
9	-	R9	Disable looping R9 Link9
10	B11	-	Put on any supported segment
11	B11	-	Disable looping B11 Link11
12	-	-	--

[0038] The hardware environment is then configured by the invention. Fig. 6 is a diagram of a configured hardware environment of that first sample topology according to the invention. Fig. 6 identifies the segments 50, and indicates 52 which segments have been enabled

[0039] As shown in Fig 6, looping links, such as Link2 to R2, Link9 to R9, and Link11 to B11 (see Fig. 3) are disabled. A device set, such as B12 that is attached to a single link is ignored.

[0040] Fig 7 is a diagram showing a second sample topology for a hardware environment according to the invention. Intelligent network agent software 62 is embedded within the hardware environment. The sample agent hardware environment 60 is a four-segment Repeater with twelve network links L1-L12. The hardware environment also is directly attached to sets of repeating devices R2, R8 and bridging devices B4, B9.

[0041] The algorithm described in the Table 1 pseudo-code is applied to this second sample topology. In this example, the preferred BridgingDevice was chosen to be the second BridgingDevice B9 because B9 has more connected links than the other BridgingDevice, B4.

[0042] Table 3 lists the actions resulting from the application of the pseudo-code algorithm of Table 1 to the second sample topology of Fig 7.

TABLE 3

Link	ConnToBridge	ConnToRptr	ACTION
1	B4	R2	Put on any supported segment
2	B4	R2	Disable looping R2 Link2
3	-	-	--
4	B4	-	Disable looping B4 Link4
5	-	-	--
6	-	-	--
7	B9	R8	Put on segCount segment, segCount++
8	B9	R8	Disable looping R8 Link8
9	B9	-	Put on segCount segment, segCount++
10	B9	-	Put on segCount segment, segCount++
11	B9	-	Put on segCount segment, segCount++
12	-	-	--

[0043] The hardware environment is then configured by the invention. Fig. 8 is a diagram of a configured hardware environment of the second sample topology according to the invention. Fig. 8 identifies the segments 70, and indicates 72 which segments have been enabled. Looping links, such as Link2 to R2, Link4 to B4, and Link8 to R8 are disabled.

[0044] In an alternative embodiment of the invention, an added enhancement can be provided for any extra BridgingDevices. In this embodiment, if the preferred BridgingDevice has fewer links than the maximum possible number of segments, one or more of the extra bridging connections that were disabled can be put on the unused segments. This results in more complete bridging communication across all of the available segments in the hardware environment.

[0045] The invention can be used to automatically configure any disabled extra repeater or BridgingLinks as redundant connections. For example, Link2 of sample topology 1 (Fig. 3) can be configured as a redundant connection for Link1 to Repeater R2. Such an intelligent agent could then monitor the main link and cause the redundant link to activate if the main link failed.

[0046] In a topology having more BridgingSet connections than segments on which the connections can be put, an extra connection can be configured as a redundant connection for use in the event of the failure of any, or of a particular, BridgingSet link. This could ensure connectivity between all devices on the network which such intelligent agents could maintain in the event of link failure, thus automatically providing connectivity equivalent to that offered by 802.1 Spanning Tree.

[0047] The invention can also be used to automatically fix connected loops. For example, extra repeater or bridging connections can be disabled or moved to a new segment. Thus, any physical loops between two or more links are fixed.

[0048] Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention.

[0049] The invention can be used with any suitable network system, including the AdvanceStack Switching Hub systems of Hewlett-Packard Company, Palo Alto, CA.

[0050] The invention uses information provided by the user, the intelligent agent, or by other applications to automatically fix network problems and/or offer users new ease-of-use features. No user action is required for such corrections. For example, this information can be used to automatically configure network links, assign bridged links to different segments to maximize the connectivity offered by the bridging, and to prevent looping. The information may additionally be used in setting up bridged links on a bridge to form a Spanning Tree with another bridge, for setting up redundant connections, and for disabling ports which are causing loops in the network.

[0051] The invention is operable in response to manual or automatic triggering. Automatic triggering includes external or internal software applications, or general policies of the agent software configuration. Thus, the configuration may be performed as desired, at pre-determined times, or in response to detected operating conditions.

[0052] Accordingly, the invention should only be limited by the Claims included below.

Claims

1. A method for automatically configuring network links in a hardware environment (10) comprising the steps of:

   assigning bridged links to different segments;
   maximizing the connectivity offered by bridging; and
   preventing looping.

2. A method for automatically configuring network links in a hardware environment (10), comprising the step of automatically setting up redundant connections.

3. A method for automatically configuring network links in a hardware environment (10), comprising the step of disabling ports which are causing loops in the network.

4. The method of Claim 1, 2 or 3, wherein an intelligent network agent is embedded in an intelligent network agent hardware environment.

5. The method of Claim 1, 2 or 3, wherein an intelligent network agent is remotely connected to said hardware environment via an out-of-band hardware control connection.

6. The method of Claim 1, 2 or 3, wherein said hardware environment is at least any one of a network bridge, switch, router, repeater, or multi-segment repeater.

7. The method of Claim 1 or 2, further comprising the step of setting up bridged links on a bridge or switch to form full connectivity across the network.

8. The method of Claim 1, further comprising the step of setting up redundant connections.

9. The method of Claim 1, further comprising the step of:
   putting ports which are causing loops in the network on different segments.

10. The method of Claim 9, further comprising the step of disabling ports which are causing loops in the network if no further segments are available.

11. The method of Claim 1, 2 or 3, wherein said method is operable in response to one of manual or automatic triggering.

12. The method of Claim 2, further comprising the step of automatically disabling ports which are causing loops in the network.

13. The method of Claim 3, further comprising the step of setting up bridged links on a bridge or switch to form a full connectivity across the network, wherein said disabled looping ports are reactivated when monitoring indicates such connectivity has been lost.

14. An apparatus for configuring network links automatically, comprising at least any one of the following:

means for assigning bridged links to different segments to maximize the connectivity offered by the bridging and to prevent looping;
means for setting up bridged links on a bridge or switch to form full connectivity across the network.
means for setting up redundant connections; and
means for disabling ports which are causing loops in the network.

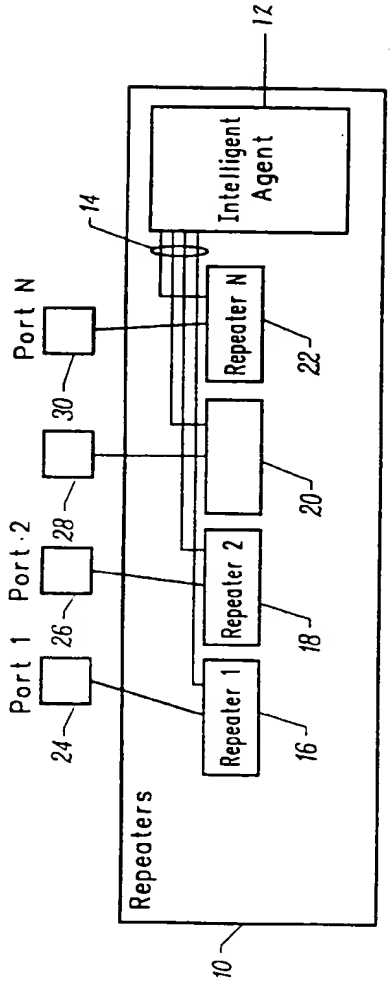

FIG. 1

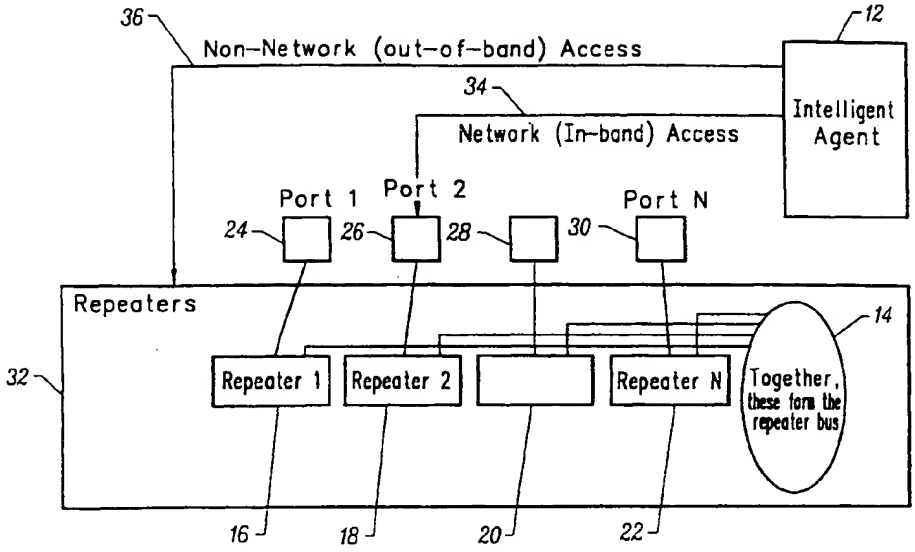

FIG. 2

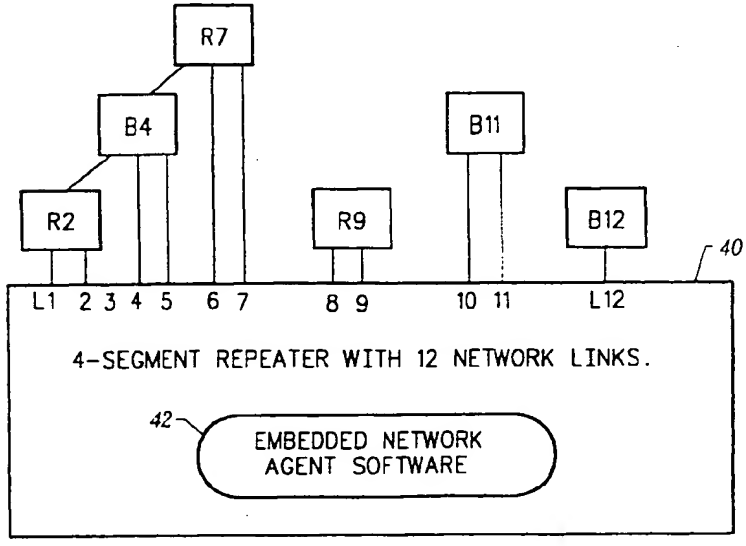

FIG. 3

	Link	1	2	3	4	5	6	7	8	9	10	11	12
44	Conn To Rate Set:	R2	R2	–	–	–	R2	R2	R2	R2	–	–	–
46	Conn To Bridge Set:	B4	B4	–	B4	B4	B4	B4	–	–	B11	B11	–

FIG. 4

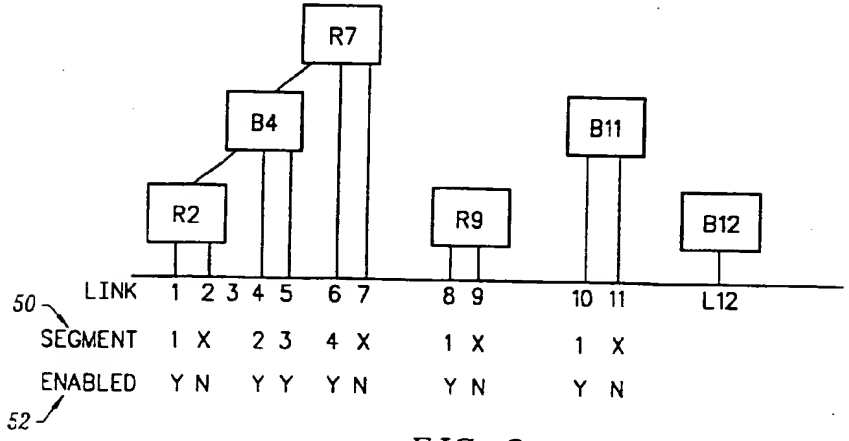

FIG. 6

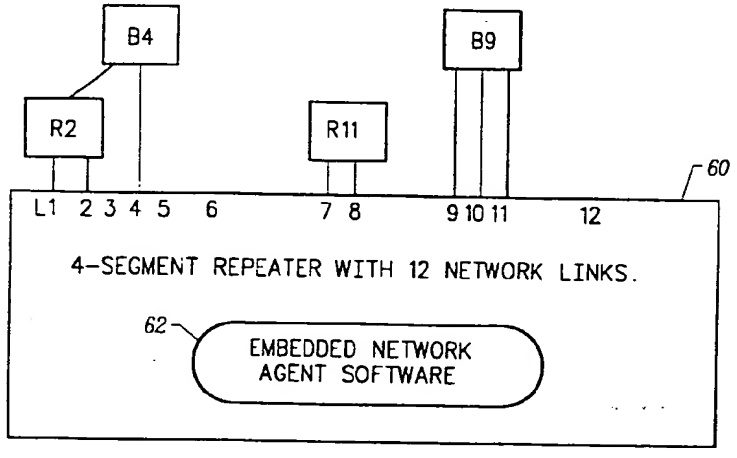

FIG. 7

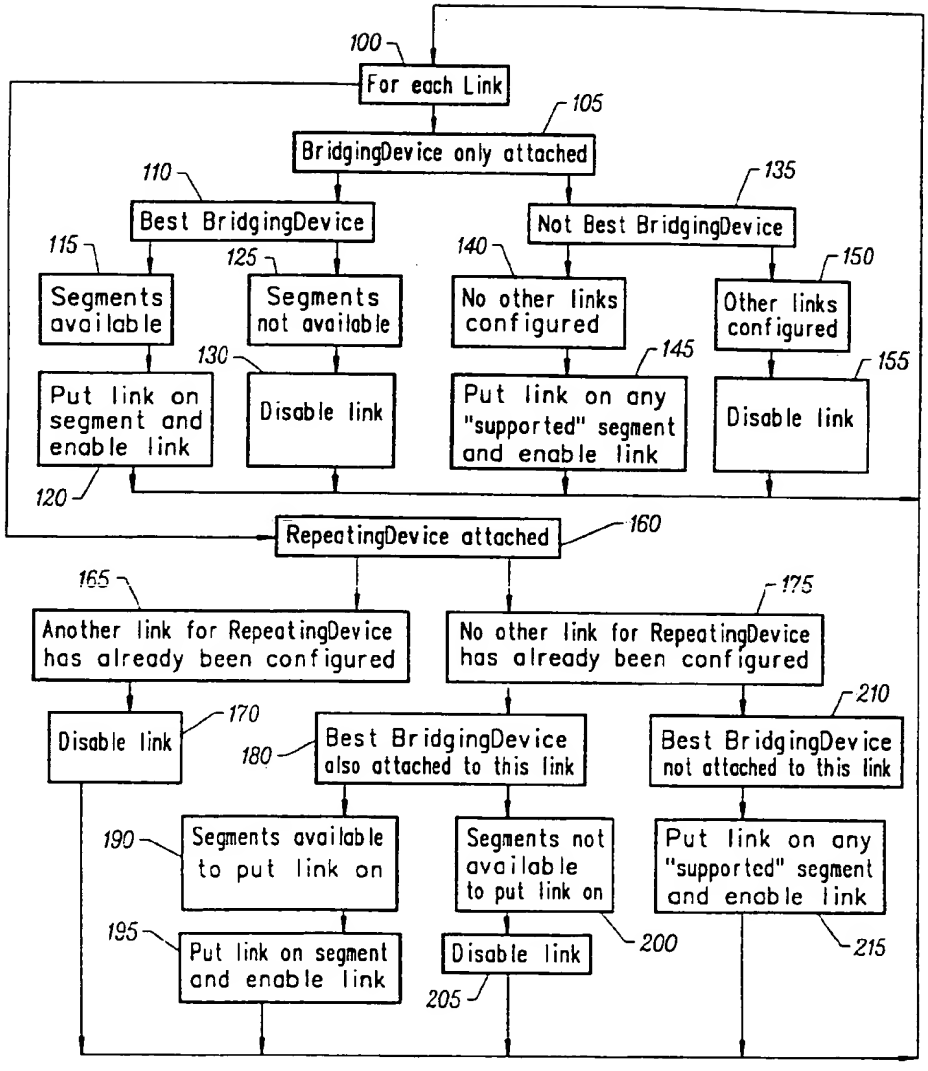

FIG. 5

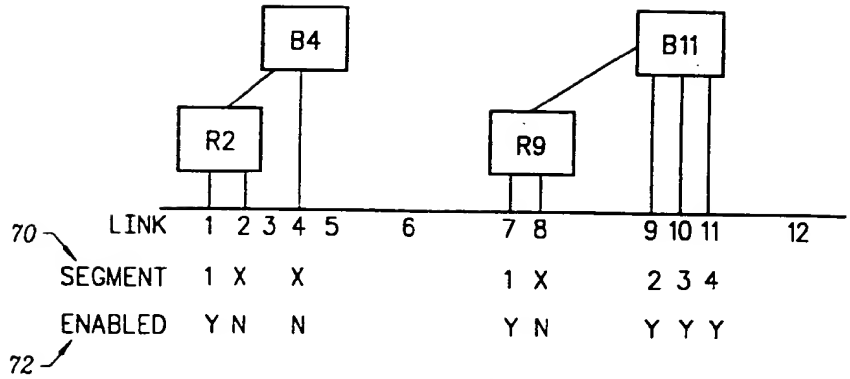

FIG. 8

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 926 863 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
18.08.1999 Bulletin 1999/33

(43) Date of publication A2:
30.06.1999 Bulletin 1999/26

(21) Application number: **98309972.2**

(22) Date of filing: **04.12.1998**

(51) Int Cl.[6]: **H04L 29/06**, H04L 12/24

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **16.12.1997 US 991943**

(71) Applicant: **Hewlett-Packard Company**
Palo Alto, California 94304 (US)

(72) Inventors:
- **Kimball, Karen E.**
  Sacramento, California 95833 (US)
- **Faulk, Robert L. Jnr.**
  Roseville, California 95747 (US)
- **Maguire, Robert M.**
  Roseville, California 95747 (US)

(74) Representative: **Powell, Stephen David et al**
WILLIAMS, POWELL & ASSOCIATES
4 St Paul's Churchyard
London EC4M 8AY (GB)

(54) Automatic configuration of network links

(57) An intelligent network agent (12) software application for multi-segment network devices (such as bridges or multi-segment repeaters) which have one or more network links automatically configures the network links. The configuration includes: assigning bridged links to different segments (to maximize the connectivity offered by the bridging and to prevent looping); setting up bridged links on a bridge to form full connectivity across the network with another bridge; setting up redundant connections; disabling ports which are causing loops in the network, or other configuration actions. The agent (12) is preferably fully embedded in the hardware device. However, the agent can also exist remotely and communicate with other software intelligence embedded in the hardware device via either an in-band (network) or an out-of-band (non-network) hardware connection.

EP 0 926 863 A3

Printed by Jouve, 75001 PARIS (FR)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 98 30 9972

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X Y	WO 95 06989 A (CABLETRON SYSTEMS INC) 9 March 1995 * abstract * * page 7, line 31 - line 35 * * page 11, line 16 - page 12, line 20 * * page 16, line 31 - line 35 *	1,3,4,6, 7,11,12 8,14	H04L29/06 H04L12/24
Y	"AUTOMATIC RECOVERY FROM FAILURE FOR ADDRESS INDEPENDENT NETWORKS" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 31, no. 11, 1 April 1989, page 435/436 XP000069848 * the whole document *	8,14	
A	"DYNAMIC LAN RECONFIGURATION MECHANISM" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 37, no. 11, 1 November 1994, pages 407-410, XP000487282 * page 408, line 22 - page 410, line 4 *	1,4,6, 11,14	
A	EP 0 349 099 A (DIGITAL EQUIPMENT CORP) 3 January 1990 * column 3, line 22 - line 37 * * column 8, line 45 - line 52 * * column 13, line 19 - column 10, line 55 *	1,2,8,14	TECHNICAL FIELDS SEARCHED (Int.Cl.6) H04L

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
BERLIN	22 June 1999	Masche, C

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 98 30 9972

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

22-06-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9506989 A	09-03-1995	AU 675362 B	30-01-1997
		AU 7868994 A	22-03-1995
		EP 0724795 A	07-08-1996
		JP 9504913 T	13-05-1997
		US 5727157 A	10-03-1998
EP 0349099 A	03-01-1990	US 4972409 A	20-11-1990
		AU 600408 B	09-08-1990
		AU 3312089 A	25-01-1990
		CA 1319185 A	15-06-1993
		DE 68921373 D	06-04-1995
		DE 68921373 T	19-10-1995
		JP 2044940 A	14-02-1990
		US 5018137 A	21-05-1991

EPO FORM P0458

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82